# 天津大学

## 本科生毕业论文




**题目：轻量级端到端语音识别系统的实现及优化**


学　　院　智能与计算学部

专　　业　计算机科学与技术

年　　级　2019 级

姓　　名　杨亦凡

学　　号　3019234258

指导教师　Daniel Povey, 王龙标

# 独创性声明

本人声明：所呈交的毕业设计（论文），是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本毕业设计（论文）中不包含任何他人已经发表或撰写过的研究成果。对本毕业设计（论文）所涉及的研究工作做出贡献的其他个人和集体，均已在论文中作了明确的说明。本毕业设计（论文）原创性声明的法律责任由本人承担。

论文作者签名：杨亦阳

2023 年 6 月 8 日

本人声明：本毕业设计（论文）是本人指导学生完成的研究成果，已经审阅过论文的全部内容。

论文指导教师签名：王龙标

2023 年 6 月 8 日

# 摘　要

神经网络传感器（Neural Transducer）和连接时序分类（Connectionist Temporal Classification，CTC）是流行的端到端自动语音识别系统。由于它们的帧同步（Frame-synchronous）设计，空白（Blank）符号被引入以解决声学帧输入序列和输出标签序列之间的长度不匹配问题，这可能会带来冗余的计算。先前的研究通过丢弃联合训练的 CTC 所预测的空白帧来加速神经网络传感器的训练和推理。然而，这并不能保证联合训练的 CTC 能够最大限度地提高空白符号的占比。本文提出了两种新颖的正则化方法，通过约束 CTC 非空白符号的自循环（Self-loop），显式地鼓励 CTC 标记更多的空白符号，使得神经网络传感器获得更大程度的加速。

在 LibriSpeech 语料库上的实验表明，本文提出的方法在不牺牲性能的前提下，将神经网络传感器的推理速度提高了 4 倍。此外，当神经网络传感器结合外部语言模型进行解码时，能够获得更大的性能提升。值得注意的是，本文提出的正则化方法能够让神经网络传感器的跳帧率逼近理论极限，这是首个探索几乎不含空白符号的神经网络传感器可行性的工作。

**关键词：**语音识别，神经网络传感器，连接时序分类

# ABSTRACT

Neural Transducer and connectionist temporal classification (CTC) are popular end-to-end automatic speech recognition systems. Due to their frame-synchronous design, blank symbols are introduced to address the length mismatch between acoustic frames and output tokens, which might bring redundant computation. Previous studies managed to accelerate the training and inference of neural Transducers by discarding frames based on the blank symbols predicted by a co-trained CTC. However, there is no guarantee that the co-trained CTC can maximize the ratio of blank symbols. This paper proposes two novel regularization methods to explicitly encourage more blanks by constraining the self-loop of non-blank symbols in the CTC.

Experiments on LibriSpeech corpus show that our proposed method accelerates the inference of neural Transducer by 4 times without sacrificing performance. It is interesting to find that the frame reduction ratio of the neural Transducer can approach the theoretical boundary. Additionally, a further gain can be observed when decoding with external language models. To the best of our knowledge, this is the first work to explore the feasibility of neural Transducers with almost no blank symbols.

**KEY WORDS:** Speech Recognition, Neural Transducer, CTC

# 目 录

# 第一章　绪论

## 1.1 研究背景及研究目的

端到端 (End to end, E2E) 架构在自动语音识别（Automatic speech recognition，ASR）领域获得越来越多的关注。近年来，主要开发出三个具有代表性的架构，分别是连接时序分类（Connectionist Temporal Classification，CTC）[1]，神经网络传感器（neural Transducer）[2] 和基于注意力的编码器-解码器（Attention-based Encoder-Decoder，AED）[3]。在这三种架构中，CTC 和神经网络传感器具有一些共同特征，因为它们都是帧同步（Frame-synchronous）系统，每一个声学输入帧都会映射到一个或多个目标标签（Label）。相比之下，AED 采用了标签同步（Label-synchronized）解码，每一步都会生成一个有效的目标标签。由于其流式处理特性和在多项任务中的卓越性能，神经网络传感器越来越受到学术研究和工业应用的关注。然而，与 AED 模型相比，神经网络传感器的解码过程计算开销更大，因为它需要处理帧同步解码中每个声学帧所对应的输出。

由于声学帧的输入序列通常比目标标签序列要长得多，帧同步的神经网络传感器和 CTC 架构中引入了一个特殊的空白符号（Blank），来表示“不输出任何内容”。在推理过程中，大部分声学输入帧被归类为空白帧，这可能导致不必要的计算。

## 1.2 国内外发展现状

为了加快解码速度，Chen 等人[4] 使用高效的空白符号以及后处理方法，将隐马尔可夫模型和 CTC 模型的推理过程从帧同步改为标签同步，在保持性能的同时取得了极大的加速。Xu 等人[5] 在标准神经网络传感器中引入额外的空白符号，称之为大空白符，输出一个大空白符号会消耗两个或更多的声学输入帧，实现了较大的推理加速，并略微提高准确率。

此外，以往的研究已经探究了空白帧的识别以及丢弃空白帧对解码结果的影响，Chen 等人[6] 研究了 CTC 模型的尖峰后验特性（Peaky Posterior Property），发现空白帧对解码性能的贡献微不足道，丢弃空白帧不会影响解码最佳路径。类似地，Zhang 等人[7] 在神经网络传感器的解码过程中丢弃空白帧并压缩搜索空间，以加快解码速度。Tian 等人[8] 在推理过程中，基于联合训练的 CTC 生成的空白概率分布丢弃共享编码器的输出帧，从而减少通过联合网络（Joiner/Joint Network）的编码器帧数。与[8] 类似，Wang 等人[9] 在训练和推理过程中在共享编码器的中间层应用了跳帧技术，在不影响性能的前提下获得了显著的推理加速。

## 1.3 本文主要工作及贡献

以往的工作通过丢弃联合训练的 CTC 预测的空白帧，从而实现神经网络传感器的推理加速[8,9]。如果联合训练的 CTC 能够准确地将更大比例的声学输入帧归类为空白帧，那么神经网络传感器的推理速度可以进一步得到加快。为了实现这一目标，本文探索了多种方法来正则化 CTC 预测的空白概率。本文通过在 CTC 拓扑图（CTC Topology）中对非空白符号（Token）的自环加以惩罚 $\lambda$，或者通过限制 CTC 中连续重复的非空白符号的最大数量 $K$，来显式鼓励 CTC 分支标记更多的空白帧。本文证明，通过调整 $\lambda$ 或 $K$，CTC 标记的非空白声学帧的数量可以接近目标标签序列的数量。在 LibriSpeech 语料库上的实验表明，通过空白正则化的 CTC 指导的神经网络传感器比不丢弃空白帧的基线模型（Baseline）具有更低的词错误率（Word-error-rate，WER）。

总的来说，本文的贡献主要有三点：

- 本文提出了两种新颖的正则化方法来显式鼓励联合训练的 CTC 标记更多的空白帧，从而进一步加快神经网络传感器的推理速度；
- 通过应用本文提出的策略，神经网络传感器的跳帧率可以逼近理论极限；
- 实验表明，在不牺牲性能的前提下，本文提出的方法通过跳帧技术将神经网络传感器模型的推理速度相对于标准神经网络传感器模型提高了 4 倍。与具有竞争力的基线模型[9] 相比，本文实现了 1.5 倍的速度提升。

本文使用 k2[10] 框架来修改 CTC 拓扑结构以及计算损失函数。相关代码已发布于开源项目 icefall[1]。

[1] https://github.com/k2-fsa/icefall

# 第二章　端到端语音识别系统概述

## 2.1　常见的三类端到端语音识别框架

### 2.1.1　连接时序分类（Connectionist Temporal Classification，CTC）

CTC[1] 是最早的端到端语音识别框架之一，包含一个编码器和一个线性层作为解码器。为了解决声学输入帧序列与目标标签序列之间的长度不匹配问题，输出词汇表 $\mathcal{V}$ 增加了一个空白符号 $\varnothing$，表示这一帧没有标签输出。给定长度为 $T$ 的声学特征输入序列 $\boldsymbol{x} = (x_1, \cdots, x_T)$，编码器产生嵌入向量 $\boldsymbol{f} = (f_1, \cdots, f_T)$。然后，这些嵌入向量进入 CTC 解码器，生成 $T$ 个条件独立的后验概率分布 $p_1, \cdots, p_T$，与词汇表 $\mathcal{V} \cup \varnothing$ 相对应。给定长度为 $U$ 的真实标签序列 $\boldsymbol{y} = (y_1, \cdots, y_U)$，其中 $y_u \in \mathcal{V}$，CTC 的目标函数定义为 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间所有可能的对齐（Alignment）的概率之和：

$$\mathcal{L}(\boldsymbol{y}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{y})} \log p\,(\boldsymbol{\pi} \mid \boldsymbol{x}), \tag{2-1}$$

其中，$\mathcal{B}(\cdot)$ 是一个多对一的映射，用于在对齐中去除连续重复的标签和所有空白符号。图 2-1 展示了 CTC 预测的符号序列通过映射 $\mathcal{B}(\cdot)$ 得到其对应的目标序列的过程。




图 2-1　$\mathcal{B}(\cdot)$ 处理 CTC 预测的符号序列得到目标序列的过程

根据 CTC 的条件独立性假设，其目标函数公式 (2-1) 可以近似地表示为：

$$\mathcal{L}(\boldsymbol{y}) \approx \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{y})} \sum_{t=1}^{T} \log p\,(\pi_t \mid \boldsymbol{x}) \tag{2-2}$$

基于加权有限状态转换器（Weighted Finite State Transducer，WFST）的解码器可以高效地实现 CTC 及其变体的算法[11,12]。一个典型的 CTC 词图（CTC

Lattice）包含三个有限状态转换器（Finite State Transducer，FST）：

- CTC 拓扑图（CTC Topology Graph，**H**），作为映射 $\mathcal{B}(\cdot)$ 的一种实现；
- 词典图（Lexicon Graph，**L**），将词典单元（Lexicon Unit）序列映射为单词（word）；
- 稠密有限状态接收器（Dense Finite State Acceptor，**Dense.FSA**），其中权重表示声学对数概率。

CTC 词图可以通过两个步骤得到。首先，将 CTC 拓扑图 **H** 和目标序列的词典图 **L** 合并（Compose）为 HL 转换器（**HL**），其将预测的符号序列转换为单词序列。然后，将 **HL** 与表示 CTC 解码器输出的概率分布的 **Dense.FSA** 相交（Intersect），生成 CTC 词图，其中包含了目标标签序列所有有效的对齐 $\mathcal{B}^{-1}(\boldsymbol{y})$。

### 2.1.2 基于注意力的编码器-解码器（Attention-based Encoder-Decoder，AED）

AED[3] 采用注意力机制[13] 来隐式识别并建模每个输出单元相关的声学输入，避免了显式建模对齐的需求。AED 模型一次性处理整个声学输入序列，为了显式指示模型已经完成了所有输出标签的生成，输出词汇表 $\mathcal{V}$ 增加了一个句子结束符号 $\langle eos\rangle$。

如图 2-2 所示，AED 模型由编码器和基于注意力机制的解码器组成。编码器将声学输入帧序列 $\boldsymbol{x}=(x_1,\cdots,x_T)$ 编码为更高层次的表示 $\boldsymbol{h}=\left(h_1,\cdots,h'_T\right)$。解码器基于注意力机制得到关于输出词汇表 $\mathcal{V}\cup\langle eos\rangle$ 的概率分布。给定训练样本对 $(\boldsymbol{x},\boldsymbol{y})$，$\hat{\boldsymbol{y}}=(y_1,\cdots,y_U,\langle eos\rangle)$ 表示通过结尾添加句子结束符号 $\langle eos\rangle$ 扩展的输出标签序列。在基于注意力机制的解码器生成任何输出之前，使用句子开始符号 $\langle sos\rangle$ 作为第一个输入 $y_0$。AED 的目标函数定义为 $\hat{y}$ 的条件概率：

$$\mathcal{L}(\hat{\boldsymbol{y}})=p(\hat{\boldsymbol{y}}\mid\boldsymbol{x})=\prod_{i=1}^{U+1}p(y_i\mid y_{i-1},\cdots,y_0=\langle sos\rangle,c_i)=\prod_{i=1}^{U+1}p(y_i\mid s_i,c_i), \tag{2-3}$$

其中，$c_i$ 表示上下文向量（Context Vector），是对编码器输出 $h_1,\cdots,h'_T$ 的线性混合值；$s_i$ 表示在输出前序标签序列之后的解码器状态，是通过前一个时间步的上下文向量 $c_{i-1}$ 和输出标签 $y_{i-1}$ 更新解码器状态 $s_{i-1}$ 得到的：

$$s_i=\mathrm{Decoder}\left(c_{i-1},s_{i-1},y_{i-1}\right) \tag{2-4}$$

在每个时间步 $i$，注意力机制生成一个上下文向量 $c_i$，其包含生成下一个标签所需的声学信息。注意力模型是基于内容的，其将当前解码器状态 $s_i$ 的内容与当前时间步的编码器输出 $h_u$ 的内容进行匹配，生成注意力向量 $\alpha_i$，$\alpha_i$ 用于加权融合编码器输出 $\boldsymbol{h}$ 以创建 $c_i$。

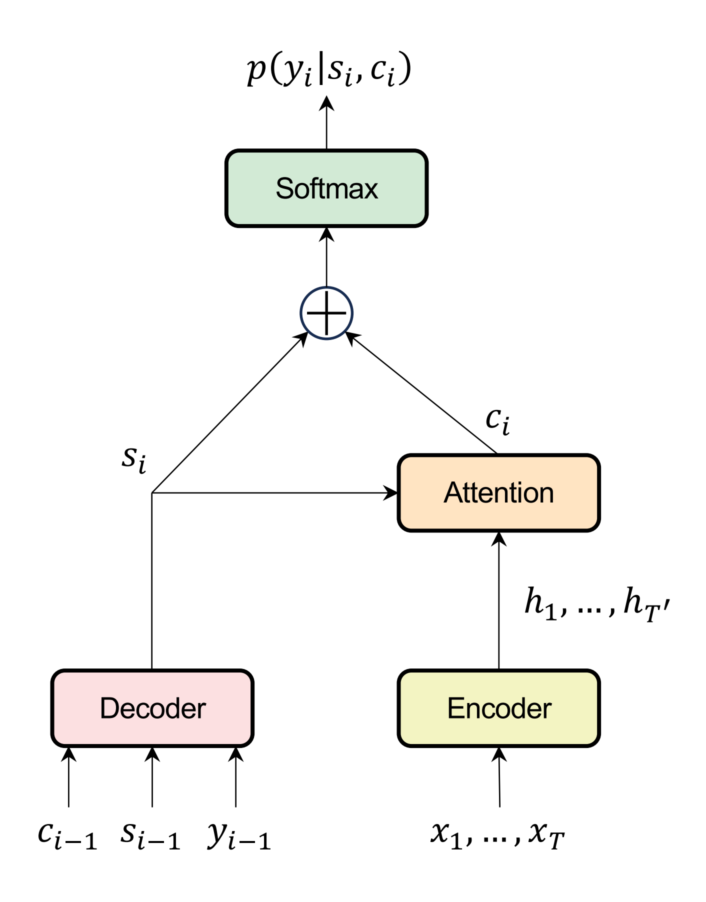



图 2-2　　基于注意力的编码器-解码器模型

具体而言，在每个解码时间步 $i$，注意力机制使用向量 $h_u$ 和 $s_i$ 为每个时间步 $u$ 计算一个标量能量 $e_{i,u}$。$e_{i,u}$ 经过 Softmax 函数转换为时间步上的概率分布 $\alpha_i$。以 $\alpha_i$ 为权重，线性混合每个时间步的编码器输出 $h_u$ 来创建上下文向量 $c_i$：

$$e_{i,u} = \langle \mathrm{MLP}_1(s_i), \mathrm{MLP}_2(h_u) \rangle \tag{2-5}$$

$$\alpha_{i,u} = \frac{\exp(e_{i,u})}{\sum_u \exp(e_{i,u})} \tag{2-6}$$

$$c_i = \sum_u \alpha_{i,u} h_u, \tag{2-7}$$

其中 $\mathrm{MLP}_1$ 和 $\mathrm{MLP}_2$ 均为多层感知器（Multilayer Perceptron，MLP）。

### 2.1.3　神经网络传感器（Neural Transducer）

神经网络传感器[2] 的提出旨在解决 CTC 的条件独立性假设所带来的问题，神经网络传感器的输出概率 $y_u$ 是以先前所有的标签 $\mathbf{y}_{\leq u-1}$ 作为条件而确定的。解码器的功能类似于语言模型，总是接收先前输出的标签作为输入。联合网络通过融合声学嵌入和文本嵌入来定义在输出 $u-1$ 个前序标签之后，在 $t$ 时间输出标签 $k$ 的概率 $p(k \mid t, u)$。类似于 CTC，神经网络传感器也定义目标函数为所有可

能的对齐的概率之和：

$$\mathcal{L}(\boldsymbol{y}) = \sum_{\boldsymbol{\pi} \in \mathcal{A}^{-1}(\boldsymbol{y})} \log p(\boldsymbol{\pi} \mid \boldsymbol{x}), \tag{2-8}$$

其中，$\mathcal{A}(\cdot)$ 是将对齐中的空白符号移除的多对一映射。

值得注意的是，CTC 和神经网络传感器中的空白符号在分隔连续符号以及对齐声学帧输入序列与目标标签序列方面具有非常相似的功能。但是，这两种类型的空白符号仍然存在一些细微的差别。具体而言，在处理具有相似声学信息的连续帧时，CTC 的对齐中允许一个目标标签对应输出序列中多个重复的符号，两个连续的相同标签需要至少一个空白符号将其隔开。而在神经网络传感器中，每个目标标签只对应输出序列中的一个符号，在对齐中其余的输出符号都是空白符号。有理由相信，这两个模型的空白符号预测是高度相关的，并有很好的同步性。

## 2.2 主流的端到端语音识别开源工具包

随着端到端自动语音识别系统引起了越来越多的研究兴趣，各种强大的开源工具包随之被开发，来普及基于 E2E 的 ASR 模型的使用，例如基于 K2[10] 的 icefall、Kaldi[14]、ESPnet[15]、WeNet[16,17] 和 Fairseq[18]。

由于本文基于 k2 框架实现，在此只讨论 k2。

### 2.2.1 k2 概述

K2 将有限状态自动机（Finite State Automaton，FSA）和有限状态转换器（Finite State Transducer，FST）算法集成到基于自动求导（Autograd）的机器学习工具包中，例如 PyTorch。k2 支持 CPU 和 CUDA，它能同时处理一批（Batch）的 FST。k2 可以用于计算 CTC 损失函数，网格无关的最大互信息（Lattice-free Maximum Mutual Information，LF-MMI）[19] 损失函数，并能用于 ASR 的解码。

k2 中的 FSA/FST 具有以下的特点：

- 只有一个起始结点（Start State）；
- 只有一个终结点（Final State）；
- 起始结点以 0 为编号；
- 其它结点的编号都大于 0；
- 终结点的编号最大；
- 进入到终结点的弧必须以 -1 作为输入标签并且分数是 0，如果有输出标签那么必须也是 -1；
- 不进入到终结点的弧不能以 -1 作为输入标签，如果有输出标签那么也不能是 -1；

- 结点不具有分数；
- 所有的分数都在弧上；
- 分数表示概率的 log 值。

### 2.2.2 基于 k2 实现 CTC 相关算法

本小节中，将以目标标签序列“AB”为例，通过代码和可视化加权有限状态转换器，展示构建 **CTC Lattice$_{\mathbf{AB}}$** 的过程，以及前向传播和反向传播的过程。

此例中，规定词汇表 $\mathcal{V}_{\mathcal{AB}}$ = [”$A$”,”$B$”]，并引入空白符号“blk”。创建 CTC 拓扑图的代码如下，其创建的 $\mathbf{H_{AB}}$ 如图 2-3 所示。

```
import k2

isym = k2.SymbolTable.from_str("""
blk 0
A 1
B 2
""")

osym = k2.SymbolTable.from_str("""
A 1
B 2
""")

H = k2.ctc_topo(max_token=2, modified=False)
H.labels_sym = isym
H.aux_labels_sym = osym
H.draw("H.pdf")
```

规定建模单元为字母，目标标签序列“AB”编码为“A”和“B”，创建词典图的代码如下，其创建的 $\mathbf{L_{AB}}$ 如图 2-4 所示。

```
L = k2.linear_fsa([1,2])
L.labels_sym = isym
L.aux_labels_sym = osym
L.draw("L.pdf")
```

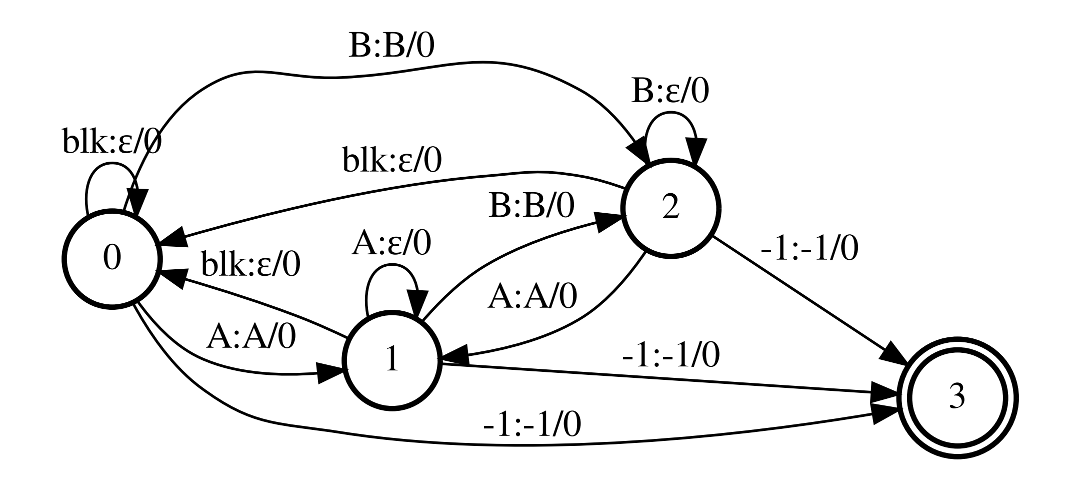



图 2-3　　表示词典为“AB”的 CTC 拓扑图

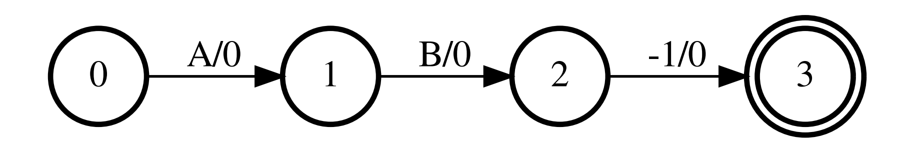



图 2-4　　表示目标标签序列“AB”的词典图

将 $\mathbf{H_{AB}}$ 与 $\mathbf{L_{AB}}$ 合并可以得到 $\mathbf{HL_{AB}}$，其将预测的符号序列转换为单词序列，包含了所有能够转换为目标序列“AB”的路径。代码如下，其创建的 $\mathbf{HL_{AB}}$ 如图 2-5 所示。

```
HL = k2.compose(H, L)
HL.draw("HL.pdf")
```

假设 CTC 解码器的输出包括三帧，其预测的概率分布如表 2-1 所示。创建其对应的稠密有限状态接收器的代码如下，其创建的 $\mathbf{Dense.FSA_{AB}}$ 如图 2-6 所示。

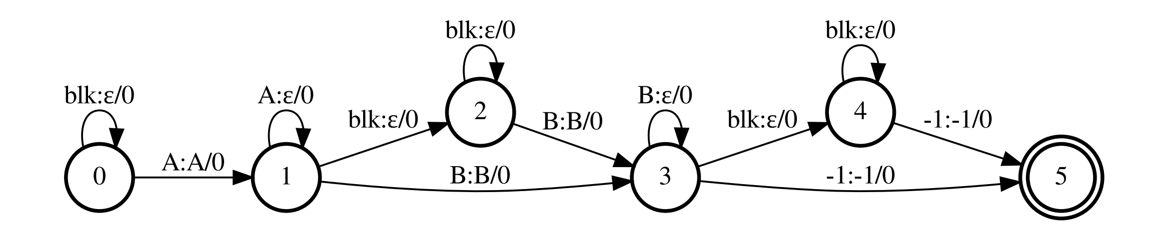



图 2-5　　目标序列“AB”对应的 HL 转换器

```
import torch

nnet_output = torch.log(torch.tensor(
    [[[0.6,0.3,0.1],[0.25,0.6,0.15],[0.25,0.15,0.6]]],
))
nnet_output.requires_grad_(True)
supervision_segments = torch.tensor([[0, 0, 3]]).to(torch.int32)
dense_fsa = k2.DenseFsaVec(
    torch.log(nnet_output),
    supervision_segments,
    allow_truncate=0,
)
dense_fsa_vec = k2.convert_dense_to_fsa_vec(dense_fsa)
dense_fsa_vec[0].draw("dense_fsa.pdf")
```

表 2-1　　CTC 解码器输出的概率分布

| frame | blk | A | B |
|---|---|---|---|
| $p_0$ | log(0.60) = -0.5108 | log(0.30) = -1.2040 | log(0.10) = -2.3026 |
| $p_1$ | log(0.25) = -1.3863 | log(0.60) = -0.5108 | log(0.15) = -1.8971 |
| $p_2$ | log(0.25) = -1.3863 | log(0.15) = -1.8971 | log(0.60) = -0.5108 |

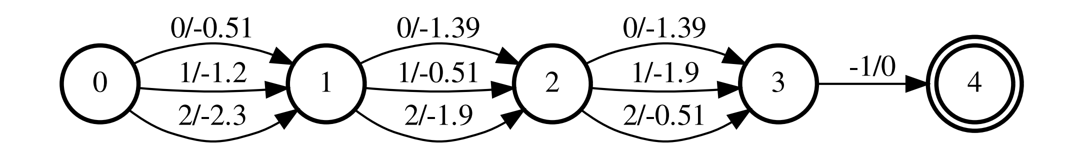



图 2-6　　CTC 解码器的输出对应的稠密有限状态接收器

将 $\mathbf{HL_{AB}}$ 与 $\mathbf{Dense.FSA_{AB}}$ 相交（Intersect），即可生成 $\mathbf{CTC\ Lattice_{AB}}$，其中包含了目标标签序列“AB”所有有效的对齐 $\mathcal{B}^{-1}(\boldsymbol{y})$。代码如下，其创建的 $\mathbf{CTC\ Lattice_{AB}}$ 如图 2-7 所示。

```
ctc_lattice = k2.intersect_dense(
    HL,
    dense_fsa,
    20.0,
)
ctc_lattice[0].draw("ctc_lattice.pdf")
```

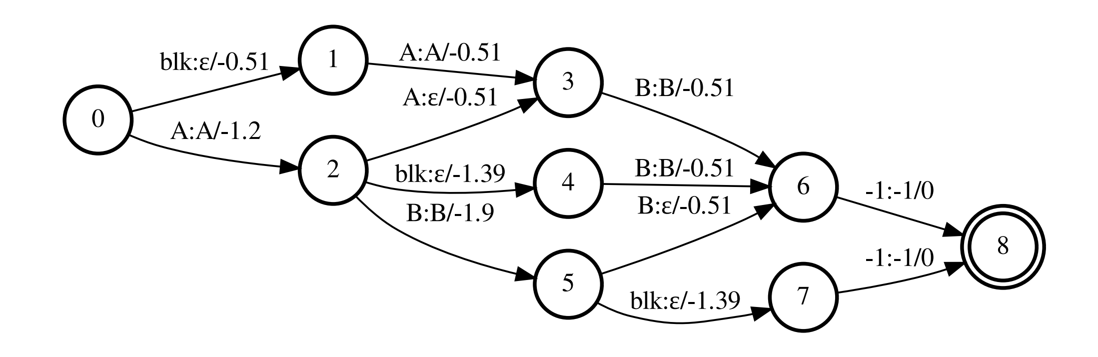



图 2-7　　目标序列“AB”对应的 CTC 词图

与传统的 CTC 中使用前向-后向算法[1] 不同，本文使用一种可微分的动态规划方法[1] 来计算 **CTC Lattice** 的总分数，进而优化 CTC 的目标函数公式 (2-2)。代码如下：

```
ctc_object_scores = ctc_lattice.get_tot_scores(
    log_semiring=True, use_double_scores=True
)
print(ctc_object_scores)
# tensor([-0.8983], dtype=torch.float64)
```

此例中使用对数半环（Log Semiring），前向传播过程的数学推导如下：

$$
\begin{aligned}
s_1 &= arc_{01} = -0.5108 \\
s_2 &= arc_{02} = -1.2040 \\
s_3 &= \log\left(e^{s_1+arc_{13}} + e^{s_2+arc_{23}}\right) = -0.6162 \\
s_4 &= s_2 + arc_{24} = -2.5903 \\
s_5 &= s_2 + arc_{25} = -3.1011 \\
s_6 &= \log\left(e^{s_3+arc_{36}} + e^{s_4+arc_{46}} + e^{s_5+arc_{56}}\right) = -0.9263 \\
s_7 &= s_5 + arc_{57} = -4.4874 \\
s_8 &= \log\left(e^{s_6+arc_{68}} + e^{s_7+arc_{78}}\right) = -0.8983
\end{aligned}
$$

其中，终结点的 $s_8 = ctc_object_scores = -0.8983$，即此例中 CTC 的目标函数公式 (2-2) 的值，这是目标标签序列“AB”所有有效的对齐 $\mathcal{B}^{-1}(\boldsymbol{y})$

k2 的绝大部分操作支持自动求导，反向传播计算梯度的代码如下：

```
ctc_object_scores.backward()
print(nnet_output.grad)
# tensor([[[0.5304, 0.4696, 0.0000],
         [0.1105, 0.7956, 0.0939],
         [0.0276, 0.0000, 0.9724]]])
```

[1]https://github.com/k2-fsa/k2 的 Fsa.get_tot_scores()

反向传播过程的数学推导如下：

$$ctc_object_scores = s_8 = \log\left(e^{s_6+arc_{68}} + e^{s_7+arc_{78}}\right)$$

$$\begin{aligned}\frac{\partial ctc_object_scores}{\partial arc_{57}} &= \frac{\partial ctc_object_scores}{\partial s_7}\frac{\partial s_7}{\partial arc_{57}}\\ &= \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_5+arc_{57}}{\partial arc_{57}}\\ &= \frac{e^{-4.4874}}{e^{-0.9263}+e^{-4.4874}}\\ &= 0.0276\end{aligned}$$

$$\begin{aligned}\frac{\partial ctc_object_scores}{\partial arc_{56}} &= \frac{\partial ctc_object_scores}{\partial s_6}\frac{\partial s_6}{\partial arc_{56}}\\ &= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial arc_{56}}\\ &= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_5+arc_{56}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\\ &= \frac{e^{-0.9263}}{e^{-0.9263}+e^{-4.4874}}\frac{e^{-3.1011-0.5108}}{e^{-0.6162-0.5108}+e^{-2.5903-0.5108}+e^{-3.1011-0.5108}}\\ &= 0.972377138857178 * 0.0681823876732906\\ &= 0.0663\end{aligned}$$

$$\begin{aligned}\frac{\partial ctc_object_scores}{\partial arc_{46}} &= \frac{\partial ctc_object_scores}{\partial s_6}\frac{\partial s_6}{\partial arc_{46}}\\ &= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial arc_{46}}\\ &= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_4+arc_{46}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\\ &= \frac{e^{-0.9263}}{e^{-0.9263}+e^{-4.4874}}\frac{e^{-2.5903-0.5108}}{e^{-0.6162-0.5108}+e^{-2.5903-0.5108}+e^{-3.1011-0.5108}}\\ &= 0.972377138857178 * 0.11363440101021251\\ &= 0.1105\end{aligned}$$

$$\begin{aligned}\frac{\partial ctc_object_scores}{\partial arc_{36}} &= \frac{\partial ctc_object_scores}{\partial s_6}\frac{\partial s_6}{\partial arc_{36}}\\ &= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial arc_{36}}\\ &= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_3+arc_{36}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\\ &= \frac{e^{-0.9263}}{e^{-0.9263}+e^{-4.4874}}\frac{e^{-2.5903-0.5108}}{e^{-0.6162-0.5108}+e^{-2.5903-0.5108}+e^{-3.1011-0.5108}}\\ &= 0.972377138857178 * 0.818183211316497\\ &= 0.7956\end{aligned}$$

$$\begin{aligned}\frac{\partial ctc_object_scores}{\partial arc_{25}} &= \frac{\partial ctc_object_scores}{\partial s_7}\frac{\partial s_7}{\partial arc_{25}} + \frac{\partial ctc_object_scores}{\partial s_6}\frac{\partial s_6}{\partial arc_{25}}\\ &= \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_5+arc_{57}}{\partial arc_{25}} + \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial arc_{25}}\\ &= \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_5+arc_{57}}{\partial arc_{25}} + \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_5}\frac{\partial s_5}{\partial arc_{25}}\\ &= \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_2+arc_{25}+arc_{57}}{\partial arc_{25}} + \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_5+arc_{56}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial s_2+arc_{25}}{\partial arc_{25}}\\ &= \frac{e^{-4.4874}}{e^{-0.9263}+e^{-4.4874}}\frac{\partial s_2+arc_{25}+arc_{57}}{\partial arc_{25}} + \frac{e^{-0.9263}}{e^{-0.9263}+e^{-4.4874}}\frac{e^{-3.1011-0.5108}}{e^{-0.6162-0.5108}+e^{-2.5903-0.5108}+e^{-3.1011-0.5108}}\\ &= 0.02762286114282208 + 0.06629899504620523\\ &= 0.0939\end{aligned}$$

$$
\begin{aligned}
\frac{\partial ctc_object_scores}{\partial arc_{24}} &= \frac{\partial ctc_object_scores}{\partial s_6}\frac{\partial s_6}{\partial arc_{24}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial arc_{24}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_4}\frac{\partial s_4}{\partial arc_{24}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_4}\frac{\partial s_2+arc_{24}}{\partial arc_{24}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_4+arc_{46}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}} \\
&= \frac{e^{-0.9263}}{e^{-0.9263}+e^{-4.4874}}\frac{e^{-2.5903-0.5108}}{e^{-0.6162-0.5108}+e^{-2.5903-0.5108}+e^{-3.1011-0.5108}} \\
&= 0.972377138857178 * 0.11363440101021251 \\
&= 0.1105
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial ctc_object_scores}{\partial arc_{23}} &= \frac{\partial ctc_object_scores}{\partial s_6}\frac{\partial s_6}{\partial arc_{23}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial arc_{23}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_3}\frac{\partial s_3}{\partial arc_{23}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_3}\frac{\log\left(e^{s_1+arc_{13}}+e^{s_2+arc_{23}}\right)}{\partial arc_{23}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_3+arc_{36}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{e^{s_2+arc_{23}}}{e^{s_1+arc_{13}}+e^{s_2+arc_{23}}} \\
&= \frac{e^{-0.9263}}{e^{-0.9263}+e^{-4.4874}}\frac{e^{-0.6162-0.5108}}{e^{-0.6162-0.5108}+e^{-2.5903-0.5108}+e^{-3.1011-0.5108}}\frac{e^{-1.204-0.5108}}{e^{-0.5108-0.5108}+e^{-1.204-0.5108}} \\
&= 0.972377138857178 * 0.818183211316497 * 0.3333215957833191 \\
&= 0.2652
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial ctc_object_scores}{\partial arc_{13}} &= \frac{\partial ctc_object_scores}{\partial s_6}\frac{\partial s_6}{\partial arc_{13}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial arc_{13}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_3}\frac{\partial s_3}{\partial arc_{13}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_3}\frac{\log\left(e^{s_1+arc_{13}}+e^{s_2+arc_{23}}\right)}{\partial arc_{13}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_3+arc_{36}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{e^{s_1+arc_{13}}}{e^{s_1+arc_{13}}+e^{s_2+arc_{23}}} \\
&= \frac{e^{-0.9263}}{e^{-0.9263}+e^{-4.4874}}\frac{e^{-0.6162-0.5108}}{e^{-0.6162-0.5108}+e^{-2.5903-0.5108}+e^{-3.1011-0.5108}}\frac{e^{-0.5108-0.5108}}{e^{-0.5108-0.5108}+e^{-1.204-0.5108}} \\
&= 0.972377138857178 * 0.818183211316497 * 0.666678404216681 \\
&= 0.5304
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial ctc_object_scores}{\partial arc_{01}} &= \frac{\partial ctc_object_scores}{\partial s_6}\frac{\partial s_6}{\partial arc_{01}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial arc_{01}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_3}\frac{\partial s_3}{\partial arc_{01}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_3}\frac{\partial s_3}{\partial s_1}\frac{\partial s_1}{\partial arc_{01}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_3}\frac{\log\left(e^{s_1+arc_{13}}+e^{s_2+arc_{23}}\right)}{\partial s_1}\frac{\partial s_1}{\partial arc_{01}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_3}\frac{\log\left(e^{s_1+arc_{13}}+e^{s_2+arc_{23}}\right)}{\partial s_1}\frac{\partial s_0+arc_{01}}{\partial arc_{01}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_3+arc_{36}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{e^{s_1+arc_{01}}}{e^{s_1+arc_{13}}+e^{s_2+arc_{23}}} \\
&= \frac{e^{-0.9263}}{e^{-0.9263}+e^{-4.4874}}\frac{e^{-0.6162-0.5108}}{e^{-0.6162-0.5108}+e^{-2.5903-0.5108}+e^{-3.1011-0.5108}}\frac{e^{-0.5108-0.5108}}{e^{-0.5108-0.5108}+e^{-1.204-0.5108}} \\
&= 0.972377138857178 * 0.818183211316497 * 0.666678404216681 \\
&= 0.5304
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial ctc_object_scores}{\partial arc_{02}} &= \frac{\partial ctc_object_scores}{\partial s_6}\frac{\partial s_6}{\partial arc_{02}} + \frac{\partial ctc_object_scores}{\partial s_7}\frac{\partial s_7}{\partial arc_{02}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial arc_{02}} + \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_7}{\partial arc_{02}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_3}\frac{\partial s_3}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_4}\frac{\partial s_4}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial \log\left(e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}\right)}{\partial s_5}\frac{\partial s_5}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_7}{\partial arc_{02}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_3+arc_{36}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial s_3}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_4+arc_{46}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial s_4}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_5+arc_{56}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial s_5}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_7}{\partial arc_{02}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_3+arc_{36}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial \log\left(e^{s_1+arc_{13}}+e^{s_2+arc_{23}}\right)}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_4+arc_{46}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial s_2+arc_{24}}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_5+arc_{56}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial s_2+arc_{25}}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_7}{\partial arc_{02}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_3+arc_{36}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial \log\left(e^{s_1+arc_{13}}+e^{s_2+arc_{23}}\right)}{\partial s_2}\frac{\partial s_2}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_4+arc_{46}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial s_2+arc_{24}}{\partial s_2}\frac{\partial s_2}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_5+arc_{56}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial s_2+arc_{25}}{\partial s_2}\frac{\partial s_2}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_7}{\partial arc_{02}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_3+arc_{36}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{e^{s_2+arc_{23}}}{e^{s_1+arc_{13}}+e^{s_2+arc_{23}}}\frac{\partial s_0+arc_{02}}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_4+arc_{46}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial s_2+arc_{24}}{\partial s_2}\frac{\partial s_0+arc_{02}}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_5+arc_{56}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{\partial s_2+arc_{25}}{\partial s_2}\frac{\partial s_0+arc_{02}}{\partial arc_{02}}+ \\
&\quad \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_7}{\partial arc_{02}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_3+arc_{36}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{e^{s_2+arc_{23}}}{e^{s_1+arc_{13}}+e^{s_2+arc_{23}}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_4+arc_{46}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_5+arc_{56}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}+ \\
&\quad \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_7}{\partial s_5}\frac{\partial s_5}{\partial s_2}\frac{\partial s_2}{\partial arc_{02}} \\
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_3+arc_{36}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{e^{s_2+arc_{23}}}{e^{s_1+arc_{13}}+e^{s_2+arc_{23}}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_4+arc_{46}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}+ \\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_5+arc_{56}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}+ \\
&\quad \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{\partial s_5+arc_{57}}{\partial s_5}\frac{\partial s_2+arc_{25}}{\partial s_2}\frac{\partial s_0+arc_{02}}{\partial arc_{02}}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_3+arc_{36}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}\frac{e^{s_2+arc_{23}}}{e^{s_1+arc_{13}}+e^{s_2+arc_{23}}}+\\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_4+arc_{46}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}+\\
&\quad \frac{e^{s_6+arc_{68}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\frac{e^{s_5+arc_{56}}}{e^{s_3+arc_{36}}+e^{s_4+arc_{46}}+e^{s_5+arc_{56}}}+\\
&\quad \frac{e^{s_7+arc_{78}}}{e^{s_6+arc_{68}}+e^{s_7+arc_{78}}}\\
&= \frac{e^{-0.9263}}{e^{-0.9263}+e^{-4.4874}}\frac{e^{-0.6162-0.5108}}{e^{-0.6162-0.5108}+e^{-2.5903-0.5108}+e^{-3.1011-0.5108}}\frac{e^{-1.204-0.5108}}{e^{-0.5108-0.5108}+e^{-1.204-0.5108}}+\\
&\quad \frac{e^{-0.9263}}{e^{-0.9263}+e^{-4.4874}}\frac{e^{-2.5903-0.5108}}{e^{-0.6162-0.5108}+e^{-2.5903-0.5108}+e^{-3.1011-0.5108}}+\\
&\quad \frac{e^{-0.9263}}{e^{-0.9263}+e^{-4.4874}}\frac{e^{-3.1011-0.5108}}{e^{-0.6162-0.5108}+e^{-2.5903-0.5108}+e^{-3.1011-0.5108}}+\\
&\quad \frac{e^{-4.4874}}{e^{-0.9263}+e^{-4.4874}}\\
&= 0.26518487850249195 + 0.11049549373005965 + 0.06629899504620525\\
&\quad +0.02762286114282208\\
&= 0.4696
\end{aligned}
$$

**CTC Lattice$_{\mathbf{AB}}$** 与 CTC 解码器输出之间的对应关系如图 2-8 所示，*nnet_output.grad* 即 $p_{00}, p_{01}\cdots, p_{22}$ 的梯度，与上述反向传播推导的结果存在以下关系：

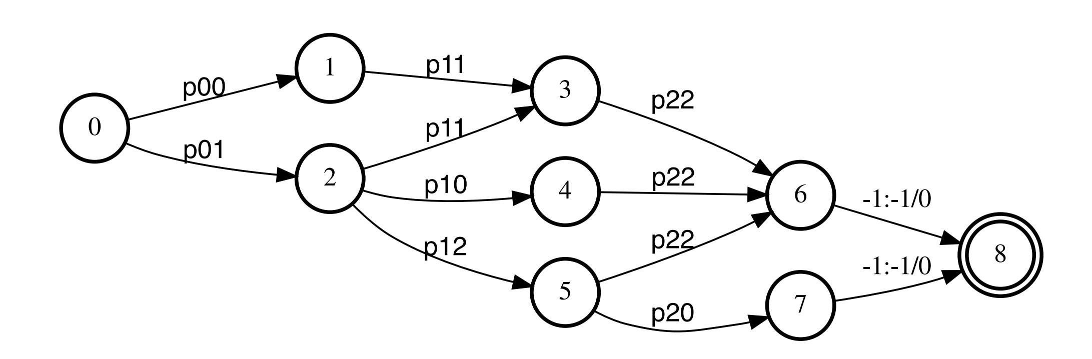



图 2-8　目标序列“AB”对应的 CTC 词图与 CTC 解码器输出的对应关系

$$
\begin{aligned}
nnet_output.grad[0][0] &= \frac{\partial ctc_object_scores}{\partial arc_{01}} = 0.5304 \\
nnet_output.grad[0][1] &= \frac{\partial ctc_object_scores}{\partial arc_{02}} = 0.4696 \\
nnet_output.grad[0][2] &= 0 \\
nnet_output.grad[1][0] &= \frac{\partial ctc_object_scores}{\partial arc_{24}} = 0.1105 \\
nnet_output.grad[1][1] &= \frac{\partial ctc_object_scores}{\partial arc_{13}} + \frac{\partial ctc_object_scores}{\partial arc_{23}} \\
&= 0.5304 + 0.2652 = 0.7956 \\
nnet_output.grad[1][2] &= \frac{\partial ctc_object_scores}{\partial arc_{25}} = 0.0939 \\
nnet_output.grad[2][0] &= \frac{\partial ctc_object_scores}{\partial arc_{57}} = 0.0276 \\
nnet_output.grad[2][1] &= 0 \\
nnet_output.grad[2][2] &= \frac{\partial ctc_object_scores}{\partial arc_{36}} + \frac{\partial ctc_object_scores}{\partial arc_{46}} + \frac{\partial ctc_object_scores}{\partial arc_{56}} \\
&= 0.7956 + 0.1105 + 0.0276 = 0.9724
\end{aligned}
$$

## 2.3 端到端语音识别系统的语言模型融合

传统的混合系统(Hybrid System)显式地学习声学模型(Acoustic Model, AM),通过贝叶斯规则，直接集成外部语言模型（External Language Model，ELM）：

$$\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}) = p_{\mathrm{AM}}(\boldsymbol{x} \mid \boldsymbol{y})\, p_{\mathrm{ELM}}(\boldsymbol{y}) \tag{2-9}$$

在端到端系统中，通过统一的模型联合学习后验概率 $p(\boldsymbol{y} \mid \boldsymbol{x})$，其隐含的声学模型和语言模型是高度相关的，这使得端到端模型在进行语言模型集成时不像混合模型那样容易。

### 2.3.1 浅融合（Shallow Fusion，SF）

浅融合[20–22] 使用在额外文本数据进行训练的循环神经网络语言模型（RNNLM），以提高语言建模能力。在解码过程中，在对数域内将神经网络传感器和循环神经网络语言模型的非空白后验概率相加:

$$\log \hat{p}(\boldsymbol{y} \mid \boldsymbol{x}) = \log p_{\mathrm{RNN-T}}(\boldsymbol{y} \mid \boldsymbol{x}) + \lambda \log p_{\mathrm{ELM}}(\boldsymbol{y}), \tag{2-10}$$

其中，$\hat{p}(\boldsymbol{y} \mid \boldsymbol{x})$ 是解码过程中使用的实际后验概率，$p_{\mathrm{RNN-T}}(\boldsymbol{y} \mid \boldsymbol{x})$ 是神经网络传感器中给定 $\boldsymbol{x}$ 时 $\boldsymbol{y}$ 的后验概率，$p_{\mathrm{ELM}}(\boldsymbol{y})$ 是由循环神经网络语言模型生成的 $\boldsymbol{y}$ 的概率。

### 2.3.2 内部语言模型估计（Internal Language Model Estimation, ILME）

神经网络传感器模型从转录文本中通过训练捕捉到的语言模型信息被认为包含在所谓的内部语言模型中（Internal language model，ILM）中。如果能够以某种方式估计神经网络传感器的内部语言模型，那么就能通过先减去内部语言

模型的后验概率 $p_{\text{ILM}}(\boldsymbol{y})$ 再加上外部语言模型的后验概率 $p_{\text{ELM}}(\boldsymbol{y})$ 来实现神经网络传感器模型的语言模型集成：

$$\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{p_{\text{RNN-T}}(\boldsymbol{x} \mid \boldsymbol{y})}{p_{\text{ILM}}(\boldsymbol{y})} p_{\text{ELM}}(\boldsymbol{y}) \tag{2-11}$$

Meng 等人[23] 中将声学隐藏状态置零，并对联合网络的输出中的非空白标签进行归一化，从而对神经网络传感器模型的内部语言模型进行估计。

### 2.3.3 密度比（Density Ratio，DR）

尽管浅融合在实践中被广泛使用，它在解码过程中将语音识别模型与循环神经网络语言模型的概率进行简单的对数线性插值，这种方法并不严格遵循贝叶斯定理。为此，密度比[24] 作为浅融合的延伸而提出，它做出如下假设：

- 源域（Source domain，即神经网络传感器训练所用的域）存在一些关于文本和音频的真实联合分布 $p_{\text{S}}(\boldsymbol{y}, \boldsymbol{x})$；
- 目标域（Target domain）存在另外一些关于文本和音频的真实联合分布 $p_{\text{T}}(\boldsymbol{y}, \boldsymbol{x})$；
- 源域的端到端模型能够合理建模 $p_{\text{S}}(\boldsymbol{y} \mid \boldsymbol{x})$；
- 独立训练的语言模型能够分别合理建模 $p_{\text{S}}(\boldsymbol{y})$ 和 $p_{\text{T}}(\boldsymbol{y})$；
- 源域和目标域具有声学一致性（Acoustically Consistent），即 $p_{\text{S}}(\boldsymbol{y} \mid \boldsymbol{x}) \approx p_{\text{T}}(\boldsymbol{y} \mid \boldsymbol{x})$；
- 目标域的后验分布是未知的 $p_{\text{T}}(\boldsymbol{y} \mid \boldsymbol{x})$。

基于上述假设，可以将目标域后验估计为：

$$\hat{p_{\text{T}}}(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{p_{\text{S}}(\boldsymbol{x})}{p_{\text{T}}(\boldsymbol{x})} \frac{p_{\text{T}}(\boldsymbol{y})}{p_{\text{S}}(\boldsymbol{y})} p_{\text{S}}(\boldsymbol{y} \mid \boldsymbol{x}) \tag{2-12}$$

公式 (2-12) 可以实现对神经网络传感器伪后验（Pseudo-posterior）的估计：

$$\log \hat{p}(\boldsymbol{y} \mid \boldsymbol{x}) = log p_{\text{RNN-T}}(\boldsymbol{y} \mid \boldsymbol{x}) + \lambda_1 \log p_{\text{ILM}}(\boldsymbol{y}) + \lambda_2 \log p_{\text{ELM}}(\boldsymbol{y}), \tag{2-13}$$

其中，$\lambda_1$ 表示源域语言模型权重,$\lambda_2$ 表示目标域语言模型权重。

### 2.3.4 低阶密度比（Low-order Density Ratio，LODR）

一些研究表明[25,26]，神经网络传感器只学习了一些低阶语言模型信息，而传统的 DR 使用了经过充分上下文训练的 RNNLM，这可能不适合于 ILM 的估计，并且可能导致语言模型集成性能下降。基于 DR 方法，Zheng 等人[27] 提出了一种低阶密度比方法（LODR），通过使用低阶弱语言模型对源领域的语言学数据进行估计。

# 第三章　空白正则化策略与跳帧

## 3.1 CTC 指导下的神经网络传感器

受 CTC 和神经网络传感器中空白符号发挥着类似作用的启发，一些先前的研究尝试利用 CTC 中的空白符号来指导并简化神经网络传感器系统的推理过程。Tian 等人[8] 提出了一种基于 CTC 分支预测的空白概率丢弃共享编码器输出帧的方法。通过减少输入联合网络来自编码器输出的帧数，可以大大提高解码速度。然而，在训练过程中没有进行跳帧。训练和推理过程中处理空白帧的方式不一致导致性能不佳。为了在训练和推理过程中实现一致的跳帧行为，Wang 等人[9] 基于联合训练的 CTC 模型预测的空白概率，在共享编码器的中间层进行跳帧。在前向传播过程中，CTC 分支计算空白概率，空白概率高于预定阈值的帧将不会参与 RNN-T 损失函数的计算。作者报道这种方法在神经网络传感器模型在训练和推理中实现了显著加速，并且不降低性能。

## 3.2 空白正则化策略

现有方法[8,9] 通过利用联合训练的 CTC 来跳过空白帧, 从而提高神经网络传感器的推理速度。然而，很少有研究探索跳过非空白帧的可行性。根据公式 (2-1) 中 $\mathcal{B}(\cdot)$ 的定义，CTC 模型不仅会删除所有空白符号，还会合并连续重复的符号。假设 CTC 分支可以将输出序列中连续重复的非空白符号对应的帧视为空白帧，那么在神经网络传感器解码时可以丢弃更多的共享编码器输出帧，从而进一步加快推理速度。受此观察的启发，本文提出了两种策略来缩小 CTC 中非空白符号后验分布的尖峰，并强制 CTC 模型输出更少的连续重复的非空白符号。

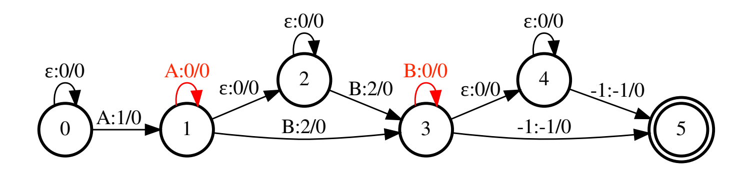
### 3.2.1 软限制（Soft Restriction）

第一个提出的策略是在训练过程中，在标准 CTC 的 HL 转换器中的所有非空白自环上引入固定惩罚项 $\lambda$（例如 0.05），如图 3.1a) 所示。这就意味着包含越多连续重复非空白符号的对齐将受到越大的惩罚，如图 3.1b) 所示，使得 CTC 模型倾向于输出具有较少连续重复非空白符号的对齐。值得注意的是，本文只在训练过程中添加这个非空白自环惩罚项 $\lambda$。通过调整惩罚项 $\lambda$ 的大小，可以控制来自 CTC 分支输出的空白帧的比例。这个策略被称为 *soft restriction*（软限制）。

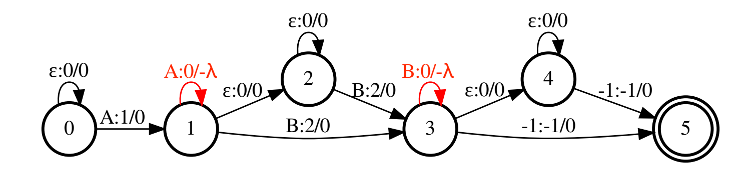
### 3.2.2 硬限制（Hard Restriction）

在 *soft restriction*（软限制）中，通过对具有连续重复非空白符号的 CTC 对齐进行惩罚，CTC 分支产生了更大比例的空白帧。然而，在训练过程中具有不同

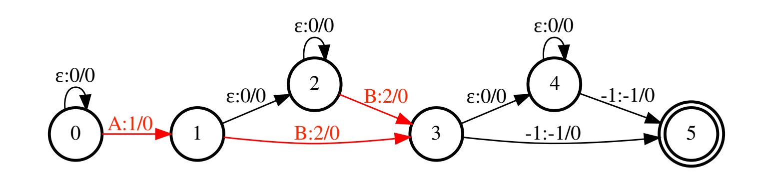
数量的重复非空白符号的对齐仍会参与优化，因为根据定义，这些都是有效的对齐。针对此，提出的另一种策略称为 *hard restriction*（硬限制），在训练过程中显式限制最大连续重复非空白符号数 $K$（包含第一个非空白符号）。图 3.1c) 和图 3.1d) 分别给出了 $K = 1$ 和 $K = 2$ 时的 HL 转换器。可以看到，超过 $K$ 个连续重复非空白符号的对齐在相应的 HL 转换器中被修剪掉了。



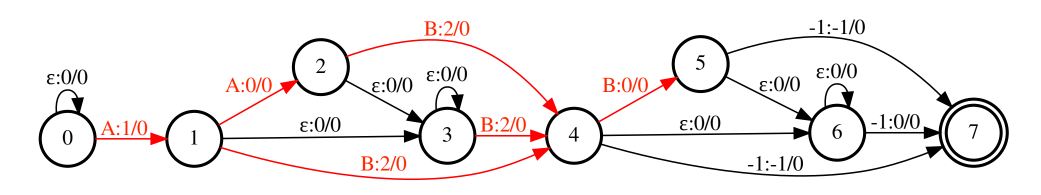

a) 标准 HL 转换器




b) 软限制的 HL 转换器，惩罚因子为 $\lambda$




c) 硬限制的 HL 转换器, $K = 1$




d) 硬限制的 HL 转换器, $K = 2$

图 3-1　目标标签序列“AB”对应的 HL 转换器的加权有限状态转换器表示，包括有/无空白正则化的情况。带有标签“**a**:**b**/**s**”的弧表示加权有限状态转换器接受输入 **a**，并以得分 **s** 输出 **b**。进入最终状态的弧的标签为“**-1**:**0**/**0**”。对于软限制，对所有非空白符号的自环加以惩罚项 $\lambda$。对于硬限制，限制连续重复非空白符号的最大数量 $K$。注意，在 $K = 1$ 的情况下，不允许连续重复非空白符号。

### 3.2.3 基于 k2 实现空白正则化的 CTC

本小节中，同样以目标标签序列“AB”为例，通过代码和可视化加权有限状态转换器，展示构建空白正则化的 CTC 的 HL 转换器的过程。

此例中，规定词汇表 $\mathcal{V}_{\mathcal{AB}} = [\text{"}A\text{"}, \text{"}B\text{"}]$，并引入空白符号“blk”。直接创建

标准的 HL 转换器的代码如下，其创建的 $\mathbf{HL_{AB}}$ 如图 3-2 所示。

```
import k2

isym = k2.SymbolTable.from_str("""
blk 0
A 1
B 2
""")

osym = k2.SymbolTable.from_str("""
A 1
B 2
""")

HL = k2.ctc_graph([[1,2]], modified=False)
HL.labels_sym = isym
HL.aux_labels_sym = osym
HL.draw("HL.pdf")
```

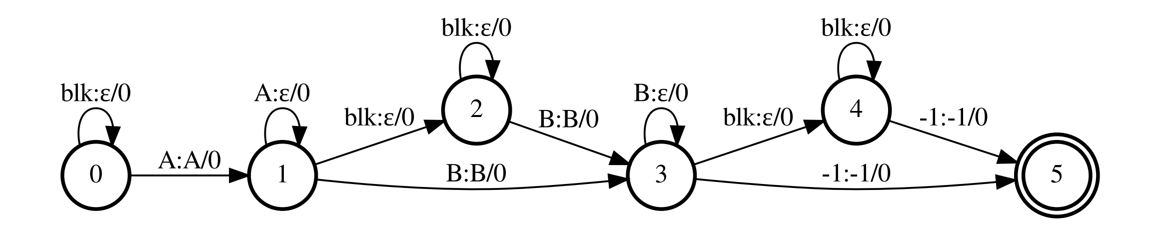



图 3-2　　表示目标序列“AB”对应的标准 HL 转换器

通过对 $\mathbf{HL_{AB}}$ 中所有非空白自环（即 $arc_{11}, arc_{33}$）加以固定惩罚项 $\lambda = 0.05$，即可得到软限制的 HL 转换器。创建软限制 $\lambda = 0.05$ 的 HL 转换器的代码如下，其创建的 $\mathbf{HL_{AB}^{\lambda=0.05}}$ 如图 3-3 所示。

```
HL_soft = k2.ctc_graph([[1,2]], modified=False)
all_self_blanks_idx = (
    HL_soft.arcs.values()[:, 0] == HL_soft.arcs.values()[:, 1]
)
blank_self_loops_idx = HL_soft.arcs.values()[:, 2] == 0
HL_soft.scores[all_self_blanks_idx] = -0.05
HL_soft.scores[blank_self_loops_idx] = 0.0
HL_soft.draw("HL_soft.pdf")
```

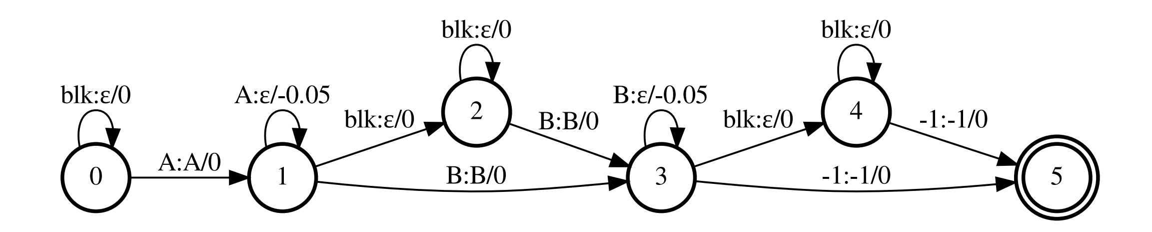



图 3-3　　表示目标序列“AB”对应的软限制的 HL 转换器, $\lambda = 0.05$

通过限制 $\mathbf{HL_{AB}}$ 中非空白符号的最大连续重复数量 $K = 2$，即可得到硬限制的 HL 转换器。创建硬限制 $K = 2$ 的 HL 转换器的代码如下，其创建的 $\mathbf{HL_{AB}^{K=2}}$ 如图 3-4 所示。

```
HL_hard = k2.fast_ctc_graph(
    [[1,2]], modified=False, max_repeat=2
)
HL_hard.draw("HL_hard2.pdf")
```

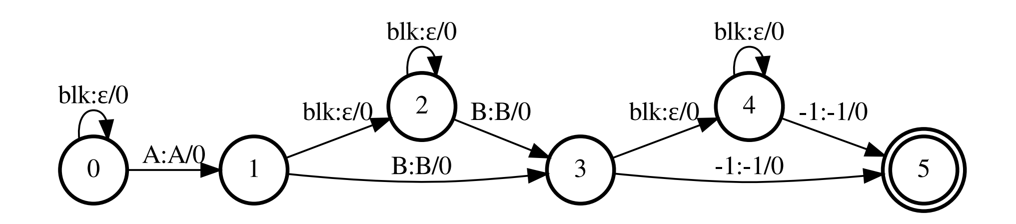



图 3-4　　表示目标序列“AB”对应的硬限制的 HL 转换器, $K = 2$

通过限制 $\mathbf{HL_{AB}}$ 中非空白符号的最大连续重复数量 $K = 1$，即可得到硬限制

的 HL 转换器。创建硬限制 $K = 1$ 的 HL 转换器的代码如下，其创建的 $\mathbf{HL}_{\mathbf{AB}}^{\mathbf{K=1}}$ 如图 3-5 所示。

```
HL_hard = k2.fast_ctc_graph(
    [[1,2]], modified=False, max_repeat=1
)
HL_hard.draw("HL_hard1.pdf")
```

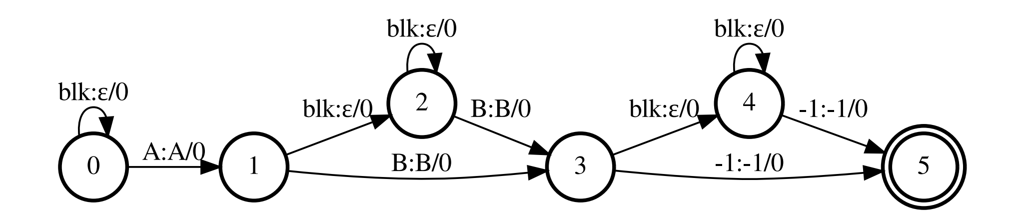



图 3-5　表示目标序列“AB”对应的硬限制的 HL 转换器, $K = 1$

## 3.3 跳帧

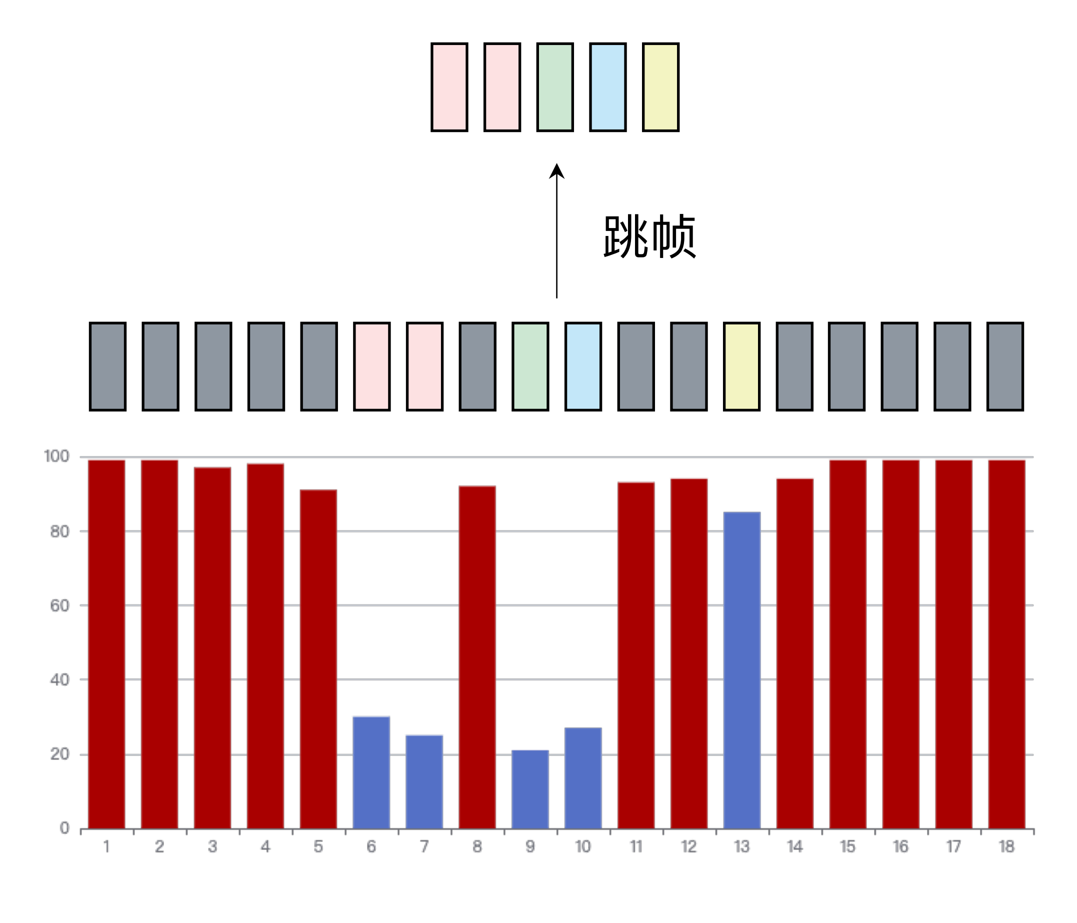



图 3-6　跳帧示意图

在训练过程中，根据联合训练的 CTC 分支预测的空白概率 $p_t^{\varnothing}$，筛选编码器

输出帧。如果 $p_t^{\varnothing}$ 超过预先设定的阈值 $\beta$（例如0.85），则第 $t$ 帧被归类为空白帧，在RNN-T损失函数计算时被丢弃。跳帧的过程如图3-6所示。被丢弃的编码器输出帧集合可以被如下描述：

$$\boldsymbol{f}_{skipped} = \{f_t \mid p_t^{\varnothing} \le \beta\} \tag{3-1}$$

在推理过程中，可以通过调整跳过空白帧的阈值 $\beta'$ 来实现跳帧率和识别准确率之间的折中。

# 第四章　实验与分析

## 4.1 实验设计

**数据集**　在实验中使用 LibriSpeech [28] 语料库，其中包含 960 小时的有声读物录音转录。使用 Lhotse[29] 进行数据准备。对训练数据应用速度扰动[30]，扰动因子为 0.9 和 1.1，来进行数据增广。在训练过程中使用 SpecAugment[31] 和噪声增强[32] 来提高模型的鲁棒性。输入特征是以窗口大小为 25 毫秒，帧移为 10 毫秒，提取的 80 通道滤波器组特征（FilterBank，FBank）。建模单元使用 500 类的字节对编码（Byte Pair Encoding，BPE）[33] 词切片（Word Pieces）。

**系统架构**　如图 4-1所示，采用了联合训练的 CTC 和神经网络传感器模型框架。共享编码器是一个维数为 512 的 12 层 Conformer[34]。神经网络传感器的解码器是一个维数为 512、上下文长度为 2 的无状态解码器（Stateless）[26]。在编码器之前有一个步长为 4 的卷积降采样模块，将帧率降低到 25 赫兹。该模型总共有 78.9M 参数。

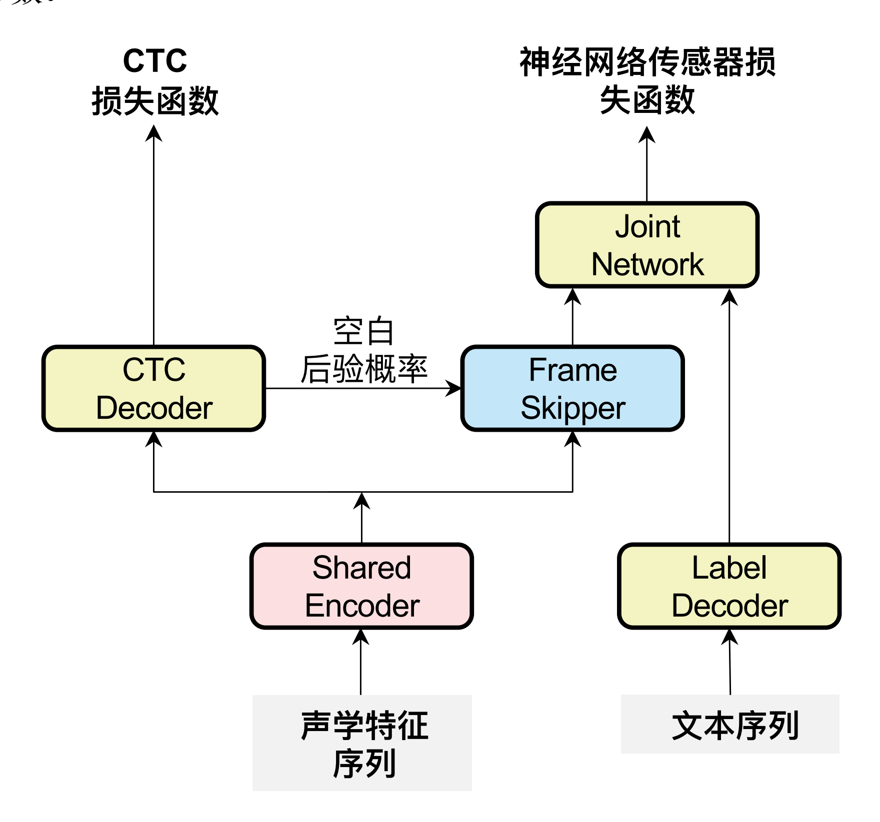



图 4-1　神经网络传感器基于联合训练的 CTC 进行跳帧的系统架构。

**训练**　采用由 CTC 损失函数[1] 的 0.2 倍与剪枝的 RNN-T 损失函数（Pruned RNN-T Loss）[35] 相加得到的损失函数。考虑到 CTC 分支在早期阶段对空白符号

的预测不准确，跳帧机制在 4000 次更新（Step）之后才会使用。本文在接下来的小节设计了一系列实验来探索超参数（$\beta, \beta', \lambda, K$）的影响。所有模型均使用 4 张 NVIDIA V100 GPU 进行训练。

**评估** 在 LibriSpeech 的测试集（test-clean 和 test-other）上对性能进行评估。除了词错误率（WER）外，还测量了跳帧率（定义为 $\frac{|f_{skipped}|}{T}$）和实时率（Real Time Factor，RTF）作为推理速度的评估指标。此外，为了评估语言模型与应用了跳帧技术的神经网络传感器的结合能力，本文还报告了使用外部语言模型（LM）进行解码的 WER。本文采用浅融合和低阶密度比分别进行了语言模型融合的研究。外部语言模型由三层长短期记忆网络（LSTM）[36] 组成，在 LibriSpeech 语言模型语料库上进行训练，低阶密度比中的低阶 n 元语言模型（n-gram）是在 LibriSpeech 的 960 小时的转录文本上训练的二元语言模型（Bi-gram）。通过在 LibrSpeech 的开发集（dev-clean 和 dev-other）上进行网格搜索（Grid Search）来调整长短期记忆网络语言模型和二元语言模型在解码时的比例。

## 4.2 实验结果及分析

### 4.2.1 空白正则化策略对性能的影响

图 4-2 展示了不同策略实现的空白正则化 CTC 对相同声学输入帧的空白标记情况。从白色部分的连续程度可以看出，本文提出的正则化方法有效地激励 CTC 将非空白重复帧标记为空白帧，使得整体上深色部分占比更大，也就意味着神经网络传感器损失计算涉及的帧数越少。

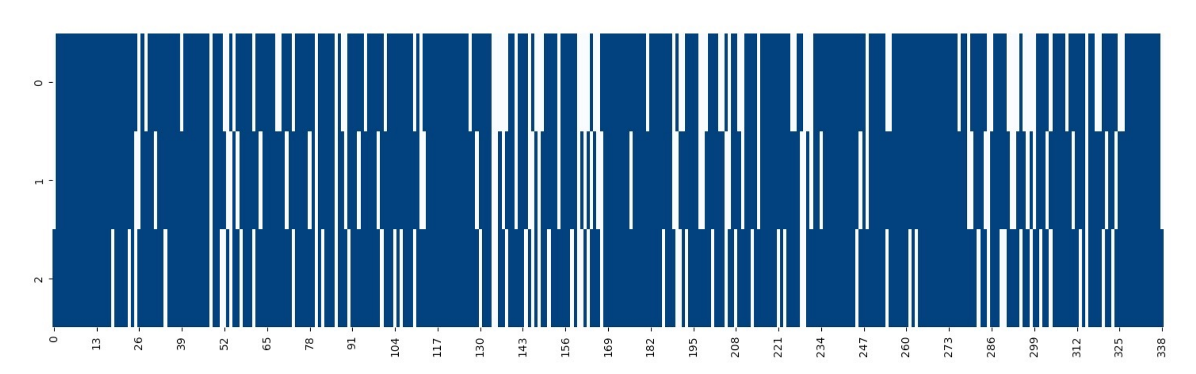



图 4-2 对比不同的策略下，空白正则化 CTC 对于相同声学输入帧的空白标记情况。深色部分表示空白帧，白色部分表示非空白帧。0 表示没有正则化的 CTC，1 表示采用软限制正则化的 CTC，2 表示采用硬限制正则化的 CTC。

应用不同策略来正则化空白符号的结果如表 4-1所示。基线模型是没有应用跳帧技术的神经网络传感器和 CTC 联合训练系统，其中 CTC 损失函数作为辅助损失函数，权重为 0.2。为了与不修改 CTC 拓扑图的现有跳帧方法进行比较，表 4-1中列出了三个在训练和推理中使用不同预设阈值进行跳帧的模型，称为

表 4-1　对比神经网络传感器在不同的策略下，使用不同的 $\beta/\lambda/K$ 的词错误率（WER，以百分比表示），跳帧率（以百分比表示）以及实时率（RTF）。

| Method | WER↓ clean | WER↓ other | Frame Reduction Ratio↑ | RTF↓ |
|---|---|---|---|---|
| Baseline | 2.45 | 5.93 | 0.00 | 0.0106 |
| Threshold | | | | |
| $\beta = 0.90$ | 2.47 | 6.07 | 65.66 | 0.0038 |
| $\beta = 0.85$ | 2.54 | 5.95 | 66.64 | 0.0036 |
| $\beta = 0.80$ | 2.59 | 6.11 | 68.34 | 0.0035 |
| Soft Restriction | | | | |
| $\lambda = 0.03$ | 2.48 | 5.91 | 74.92 | 0.0026 |
| $\lambda = 0.04$ | **2.44** | **5.88** | **75.44** | **0.0026** |
| $\lambda = 0.05$ | 2.51 | 6.00 | 75.78 | 0.0024 |
| $\lambda = 5.00$ | 2.90 | 6.89 | 78.25 | 0.0023 |
| Hard Restriction | | | | |
| $K = 2$ | 2.49 | 5.92 | 72.35 | 0.0031 |
| $K = 1$ | 2.92 | 6.96 | 78.23 | 0.0023 |

*Threshold*。表 4-1中还显示了测试集上的平均跳帧率。作为一个参考数值，本文计算了理论上可能的最大跳帧率 $\gamma_{max} = 1 - \frac{S}{T} = 78.61\%$，其中 $S$ 是测试集中输出符号序列的总长度，$T$ 是测试集中声学输入帧序列的总长度。

可以得出以下观察结果：

- 与现有方法相比，在 CTC 拓扑图上应用软限制或硬限制后实现了更大的跳帧率，表明有更大比例的帧被 CTC 分支标记为空白帧；
- 可以通过调整 $\lambda$ 或 $K$ 来实现在词错误率和实时率之间的折中。较大的惩罚 $\lambda$ 或较小的 $K$ 激励共享编码器更大力度区分空白帧和非空白帧，从而使得空白帧的预测更加可靠。通过进一步增加 $\lambda$ 至 5 或将 $K$ 减小至 1，跳帧率趋近 $\gamma_{max}$，同时保持合理的准确率；
- 本文提出的空白正则化方法在保持与不跳过空白帧的基线模型相当的甚至稍微更低的词错误率的同时，实现了 4 倍的加速。这表明在训练过程中通过适当的正则化，连续重复的非空白帧对解码结果的贡献可以忽略不计，在推理过程中丢弃这些帧不会影响词错误率。

图 4-3 可视化了跳帧率与聚合词错误率之间的关系，聚合词错误率是在 Lib-

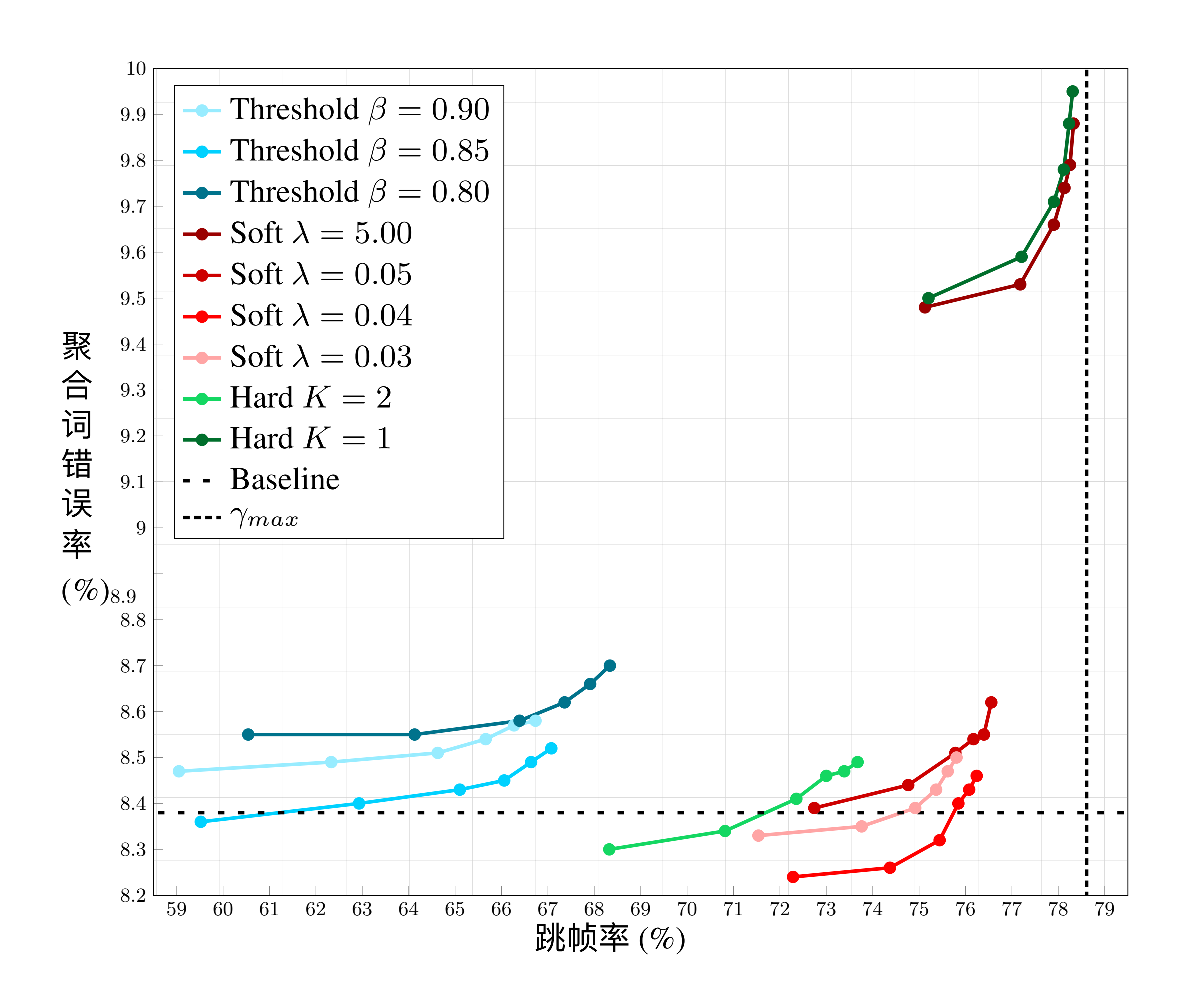



图 4-3　聚合词错误率（以百分比表示）关于跳帧率（以百分比表示）的曲线。聚合词错误率是在 LibriSpeech 语料库的 test-clean 和 test-other 子集上的词错误率之和。

riSpeech 语料库的 test-clean 和 test-other 子集上的词错误率之和。图 4-3 中还包括基线模型的聚合词错误率（水平虚线）和 $\gamma_{max}$（垂直虚线），两者作为参考线绘制。每条曲线的采样点是通过改变解码空白阈值 $\beta'$（取值为 0.8, 0.85, 0.9, 0.95, 0.99, 0.999）得到的。从图中可以看出，在解码过程中通过调整 $\beta'$ 可以实现聚合词错误率和跳帧之间的折中。较小的 $\beta'$ 能得到更多的空白帧，但同时也会增加词错误率。本文提出的方法在不牺牲准确率的前提下实现了比现有方法（图 4-3 中 Threshold-$x$）更大的跳帧率。通过调整 $\lambda$ 或 $K$，使用空白正则化的神经网络传感器甚至优于不跳过空白帧的基线模型，同时实现超过 75% 的跳帧率。

### 4.2.2　空白正则化策略对外部语言模型融合的影响

表 4-2 展示了使用外部语言模型解码的词错误率（WER），并且展示了相对于表 4-1 中不使用外部语言模型作为基线模型的相对性能提升百分比。

与基线模型相比，使用空白正则化的神经网络传感器模型在与外部语言模型结合时，无论采用浅融合或是低阶密度比，均使得词错误率相对降低得更多，

表 4-2　对比不同策略下的词错误率（WER，以百分比表示）及与外部语言模型融合的相对性能提升百分比。

| 方法 | 浅融合↓ | | 提升百分比↑ | 低阶密度比↓ | | 提升百分比↑ |
|---|---|---|---|---|---|---|
| | clean | other | | clean | other | |
| Baseline | 2.24 | 5.35 | 9.43 | 2.16 | 5.09 | 13.48 |
| Threshold | | | | | | |
| $\beta = 0.85$ | 2.23 | 5.23 | 12.13 | 2.12 | 5.05 | 15.55 |
| Soft Restriction | | | | | | |
| $\lambda = 0.04$ | 2.19 | 5.17 | 11.54 | 2.09 | 4.94 | 15.50 |
| $\lambda = 5.00$ | 2.55 | 6.16 | 11.03 | 2.44 | 5.87 | 15.12 |
| Hard Restriction | | | | | | |
| $K = 2$ | 2.19 | 5.21 | 12.01 | 2.07 | 5.01 | 15.81 |
| $K = 1$ | 2.55 | 6.10 | 12.45 | 2.47 | 5.78 | 16.50 |

这表明丢弃更多的空白帧可以更好地与外部语言模型融合。

# 第五章　结论与展望

受到 CTC 和神经网络传感器中空白符号发挥着类似作用的启发，本文提出了两种新颖的空白正则化方法，以进一步提高联合训练的 CTC 模型中预测空白符号的比例。通过在进入联合网络之前丢弃空白帧，空白正则化的 CTC 可以极大地加速神经网络传感器的推理过程。值得注意的是，本文提出的正则化方法能够让神经网络传感器的跳帧率逼近理论极限。实验证明，在空白正则化的 CTC 的指导下，神经网络传感器在推理过程中实现了 4 倍的加速，并且不牺牲性能。本文的方法在词错误率（WER）和推理实时率（RTF）之间实现了更好的权衡，优于现有方法。此外，当神经网络传感器结合外部语言模型进行解码时，能够获得更大的性能提升。

本文的工作揭示了几乎不含空白符号的神经网络传感器可行性，未来将尝试重新定义神经网络传感器的空白符号，实现更高效的端到端自回归框架。

# 参考文献


[1] Graves A, Fernández S, Gomez F J, *et al*. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks [C]. In Proc. ICML, Pittsburgh, 2006.

[2] Graves A, Mohamed A, Hinton G E. Speech recognition with deep recurrent neural networks [C]. In Proc. ICASSP, Vancouver, 2013.

[3] Chan W, Jaitly N, Le Q, *et al*. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition [C]. In Proc. ICASSP, Shanghai, 2016.

[4] 陈哲怀, 郑文露, 游永彬等. 标签同步解码算法及其在语音识别中的应用 [J]. 计算机学报, 2019, 42 (1511-1523).

[5] Xu H, Jia F, Majumdar S, *et al*. Multi-blank Transducers for Speech Recognition [C]. In Proc. ICASSP, Rhode Island, 2023.

[6] Chen Z, Deng W, Xu T, *et al*. Phone Synchronous Decoding with CTC Lattice [C]. In Proc. Interspeech, San Francisco, 2016.

[7] Zhang Y, Sun S, Ma L. Tiny Transducer: A Highly-Efficient Speech Recognition Model on Edge Devices [C]. In Proc. ICASSP, Toronto, 2021.

[8] Tian Z, Yi J, Bai Y, *et al*. FSR: Accelerating the Inference Process of Transducer-Based Models by Applying Fast-Skip Regularization [C]. In Proc. Interspeech, Brno, 2021.

[9] Wang Y, Chen Z, Zheng C, *et al*. Accelerating RNN-T Training and Inference Using CTC guidance [C]. In arXiv preprint arXiv:2210.16481, 2022.

[10] Povey D, Zelasko P, Khudanpur S. Speech recognition with next-generation kaldi (k2, lhotse, icefall) [C]. In Interspeech: tutorials, 2021.

[11] Miao Y, Gowayyed M, Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding [C]. In Proc. ASRU, Scottsdale, 2015.

[12] Laptev A, Majumdar S, Ginsburg B. CTC Variations Through New WFST Topologies [C]. In Proc. Interspeech, Incheon, 2022.

[13] Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need [C]. In Proc. NIPS, Long Beach, 2017.

[14] Povey D, Ghoshal A, Boulianne G, *et al*. The Kaldi speech recognition toolkit [C]. In Proc. ASRU, Waikoloa, 2011.

[15] Watanabe S, Hori T, Karita S, *et al*. Espnet: End-to-end speech processing toolkit [C]. In arXiv preprint arXiv:1804.00015, 2018.

[16] Yao Z, Wu D, Wang X, *et al*. WeNet: Production oriented Streaming and Non-streaming End-to-End Speech Recognition Toolkit [C]. In Proc. Interspeech, Brno, 2021.

[17] Zhang B, Wu D, Peng Z, *et al*. WeNet 2.0: More Productive End-to-End Speech Recognition Toolkit [C]. In Proc. Interspeech, Incheon, 2022.

[18] Ott M, Edunov S, Baevski A, *et al*. fairseq: A Fast, Extensible Toolkit for Sequence Modeling [C]. In Proc. NAACL, Minneapolis, 2019.

[19] Hadian H, Sameti H, Povey D, *et al*. End-to-end Speech Recognition Using Lattice-free MMI. [C]. In Proc. Interspeech, Hyderabad, 2018.

[20] Mikolov T, Karafiát M, Burget L, *et al*. Recurrent neural network based language model [C]. In Proc. Interspeech, Chiba, 2010.

[21] Chorowski J, Jaitly N. Towards better decoding and language model integration in sequence to sequence models [C]. In Proc. Interspeech, Stockholm, 2017.

[22] Kannan A, Wu Y, Nguyen P, *et al*. An Analysis of Incorporating an External Language Model into a Sequence-to-Sequence Model [C]. In Proc. ICASSP, Calgary, 2018.

[23] Meng Z, Kanda N, Gaur Y, *et al*. Internal language model training for domain-adaptive end-to-end speech recognition [C]. In Proc. ICASSP, Toronto, 2021.

[24] McDermott E, Sak H, Variani E. A density ratio approach to language model fusion in end-to-end automatic speech recognition [C]. In Proc. ASRU, Singapore, 2019.

[25] Variani E, Rybach D, Allauzen C, *et al*. Hybrid autoregressive transducer (hat) [C]. In Proc. ICASSP, Barcelona, 2020.

[26] Ghodsi M, Liu X, Apfel J, *et al*. RNN-transducer with stateless prediction network [C]. In Proc. ICASSP, Barcelona, 2020.

[27] Zheng H, An K, Ou Z, *et al*. An Empirical Study of Language Model Integration for Transducer based Speech Recognition [C]. In Proc. Interspeech, Incheon, 2022.

[28] Panayotov V, Chen G, Povey D, *et al*. Librispeech: an asr corpus based on public domain audio books [C]. In Proc. ICASSP, South Brisbane, 2015.

[29] Zelasko P, Povey D, Trmal J Y, *et al*. Lhotse: a speech data representation library for the modern deep learning ecosystem [C]. In arXiv preprint arXiv:2110.12561, 2021.

[30] Ko T, Peddinti V, Povey D, *et al*. Audio augmentation for speech recognition [C]. In Proc. Interspeech, Dresden, 2015.

[31] Park D S, Chan W, Zhang Y, *et al*. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition [C]. In Proc. Interspeech, Graz, 2019.

[32] Snyder D, Chen G, Povey D. Musan: A music, speech, and noise corpus [C]. In arXiv preprint arXiv:1510.08484, 2015.

[33] Sennrich R, Haddow B, Birch A. Neural Machine Translation of Rare Words with Subword Units [C]. In Proc. ACL, Berlin, 2016.

[34] Gulati A, Qin J, Chiu C, *et al*. Conformer: Convolution-augmented Transformer for Speech Recognition [C]. In Proc. Interspeech, Shanghai, 2020.

[35] Kuang F, Guo L, Kang W, *et al*. Pruned RNN-T for fast, memory-efficient ASR training [C]. In Proc. Interspeech, Incheon, 2022.

[36] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural computation, 1997, 9.

# 致　谢

行文至此，学至尾声。己亥而始，癸亥而终，初入电信，后习计科，浅尝数学，止于皮毛。师长学长，不吝赐教，桃李不言, 下自成蹊。承蒙众爱，不甚感激。

在此，感谢我的毕业论文的指导老师王龙标，感谢提供修改意见的刘雪莉和王晓宝老师。感谢智算学部的李克秋、魏继增、冯伟、曲雯毓、万亮、林迪、尚凡华、熊德意、吴虎统、许光全、王立、王博、王旗龙、陈世展、陈俊洁、毕重科、郝建业、张怡、李罡、李森、李春、李幼萌、张小旺、金弟、刘爽、刘健、孙越恒、曲日、苏冉、饶国政、汤善江老师。感谢我的博士导师陈谐和俞凯。感谢我在小米实习的导师 Daniel Povey 和匡方军，和我的同事们杨笑宇、郭理勇、姚增伟、康魏、林珑。最后，感谢我的宝贝袁芳。

# 攻读学士学位期间主要的研究成果

已发表的论文

[1] Yang Y, Yang X, Guo L, et al. Blank-regularized CTC for Frame Skipping in Neural Transducer [C]. In Proc. Interspeech, Dublin, 2023.（CCF C，第一作者，2023 05 录用）

[2] Yao Z, Kang Wei, Kuang F, et al. Delay-penalized CTC implemented based on Finite State Transducer [C]. In Proc. Interspeech, Dublin, 2023.（CCF C，第六作者，2023 05 录用）